

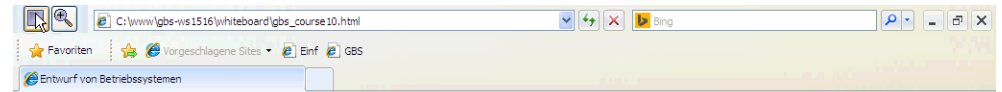
Script generated by TTT

Title: Grundlagen_Betriebssysteme (27.01.2016)

Date: Wed Jan 27 13:15:46 CET 2016

Duration: 45:41 min

Pages: 11



Der Entwurf eines BS erfordert ein ingenieurmäßiges Vorgehen. Es gibt jedoch keine wohl-definierten Vorgehensweisen, sondern nur Erfahrungen von Entwicklern bzw. Nutzern.

Einführung

Die Ziele von Betriebssystemen können sich zwischen verschiedenen Systemen unterscheiden, für Server-Systeme, für Laptops, für Smartphones oder für eingebettete Systeme.

Hauptaspekte

Probleme

Schnittstellenentwurf

Ein BS bietet eine Reihe von Diensten, die von Benutzerprozessen in Anspruch genommen werden können.

Bereitstellen der Dienste über Schnittstellen.

Leitlinien für den Entwurf

Paradigmen der Systemaufrufschnittstelle

Für die Einbindung der Systemaufrufe in Nutzerprogramme kann man zwischen den Ausführungs- und den Datenparadigmen unterscheiden.

Algorithmischer Ansatz

Ereignis-basierter Ansatz

Datenparadigma

Systemaufrufschnittstelle

Weitere Implementierungsaspekte

Trends beim Entwurf von Betriebssystemen

Generated by Targeteam



Weitere Implementierungsaspekte



Neben den Diensten und deren Bereitstellung über die Systemaufrufschnittstelle stellen die folgenden Implementierungsaspekte eine wichtige Rolle beim Entwurf von Betriebssystemen.

[Architektur](#)

[Mechanismen vs. Policies](#)

[Namensräume](#)

[Statische - Dynamische Datenstrukturen](#)

[Verbergen der Hardware](#)

[Speicherplatz vs. Laufzeit](#)

Generated by Targeteam



Architektur



Ein etablierter und erprobter Architekturansatz sind geschichtete Systeme.

Systemaufrufbehandlung		
Dateisystem 1	...	Dateisystem m
Virtueller Speicher (Paging)		
Treiber 1	...	Treiber n
Threads, Thread Scheduling, Thread Synchronisation		
Unterbrechungsbehandlung, Dispatcher, MMU		
Abstraktion der Hardware (z.B. HAL von Windows)		

Generated by Targeteam



Neben den Diensten und deren Bereitstellung über die Systemaufrufchnittstelle stellen die folgenden Implementierungsaspekte eine wichtige Rolle beim Entwurf von Betriebssystemen.

[Architektur](#)

[Mechanismen vs. Policies](#)

[Namensräume](#)

[Statische - Dynamische Datenstrukturen](#)

[Verbergen der Hardware](#)

[Speicherplatz vs. Laufzeit](#)

Generated by Targeteam



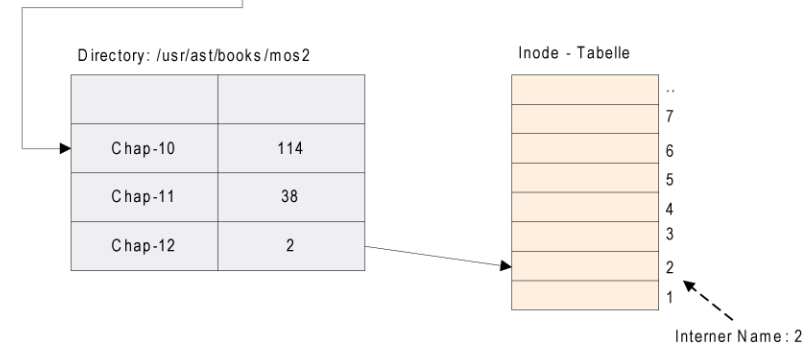
Die meisten langlebigen Datenstrukturen eines BS haben einen Namen oder einen Bezeichner zugeordnet, z.B. Login-Namen, Dateinamen, Gerätenamen, Prozess-ID etc. Namen werden auf 2 Ebenen vergeben

extern: Namen, welche Menschen verwenden (z.B. Dateiname)

intern: Namen, welche das System verwendet (z.B. inode Nummer)

Directories bilden externe Namen auf interne Namen ab.

Externer Name: /usr/ast/books/mos 2/Chap-12



Generated by Targeteam



Neben den Diensten und deren Bereitstellung über die Systemaufrufchnittstelle stellen die folgenden Implementierungsaspekte eine wichtige Rolle beim Entwurf von Betriebssystemen.

[Architektur](#)

[Mechanismen vs. Policies](#)

[Namensräume](#)

[Statische - Dynamische Datenstrukturen](#)

[Verbergen der Hardware](#)

[Speicherplatz vs. Laufzeit](#)

Generated by Targeteam



Statische Strukturen sind einfacher und schneller, jedoch weniger flexibel als dynamische Strukturen

Beispiel Prozesstabelle:

Statisch: feste Prozesstabelle mit 256 Einträgen.

Dynamisch: Prozesstabelle als verkettete Liste.

Schnelle Suche in statischer Prozesstabelle.

Statische Tabellen sind vernünftig

Wenn die Zahl der Einträge sehr gut vorausgesagt werden kann.

Generated by Targeteam



Bedingt durch den vielfältigen Einsatz der Rechner sind die jeweiligen Trends von unterschiedlicher Bedeutung

BS mit großem Adressraum

Übergang von 32 Bit zu 64 Bit-Adressräumen

Adressraum von $2 \cdot 10^{19}$ Bytes

Netze werden die Grundlage für BS

Zugriff auf Web ist vollständig und nahtlos in BS integriert

Bessere Unterstützung von parallelen und verteilten Systemen.

BS für batteriebetriebene Computer (PDAs, Smart Phone, etc.)

Energiemanagement; Adaption von BS und Anwendungen je nach Energieversorgung

Trend hin zu "Green Computing".

BS für eingebettete Systeme

Generated by Targeteam

Die Bezeichnung "**Eingebettetes System**" bezieht sich auf die Nutzung von Computer Hardware und Software in einem Produkt. Unterscheidung nach Anwendungsdomäne

Automobil, z.B. Zündung, Motorkontrolle, Bremssystem

Consumer Elektronik, z.B. Set-top Box, Küchengeräte, Kamera, Smartphone

Medizinbereich, z.B. Dialysegerät, Infusionspumpe

Bürobereich, z.B. Faxgerät, Drucker

industrielle Nutzung, z.B. Roboter und Kontrollsysteme für Produktion

Eingebettete Systeme sind eng gekoppelt mit dem Produkt, wobei die Anforderungen sehr variieren können

Größe des Produkts, in das es eingebettet ist.

Lebensdauer.

Nutzungsumgebung.

Realzeitverhalten.

Es wurden eine Vielzahl von speziellen Betriebssystemen für eingebettete Systeme realisiert

für mobile Endgeräte: Symbian OS, Windows CE oder **Android** von Google.

Android Systemarchitektur basiert auf Linux Kernel (Treiber, Speicher-/Prozessmanagement); darauf existieren Bibliotheken (z.B. für Graphik), eine Laufzeitumgebung und ein Anwendungsframework.

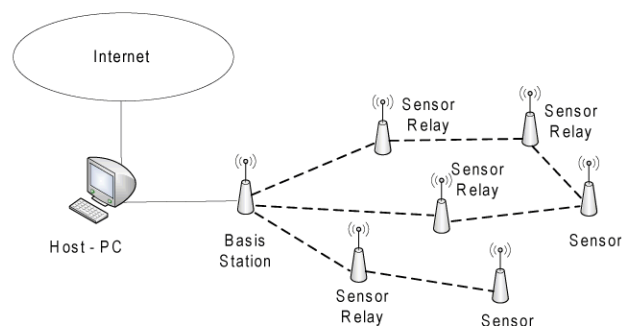
TinyOS

Generated by Targeteam



für drahtlose Sensornetze: **TinyOS**, entwickelt von der UC Berkeley

zentraler Teil des BS ist sehr klein, ca. 400 Bytes Code und Datenspeicher



TinyOS hat keinen BS-Kern und keinen Speicherschutzmechanismus.

komponenten-basierter Ansatz und ereignis-basiertes Ausführungsmodell.

geschichtete Anordnung der Komponenten; zwischen Komponenten bestehen Aufrufbeziehungen; elementare Komponenten kapseln Hardware Komponenten (z.B. Timer).

Anwendungen werden durch Verknüpfung von Komponenten realisiert.

2 Arten von Komponenten: a) Module = implementieren Funktionen, und b) Konfigurationen = beschreiben die Verknüpfung von Komponenten.

eine Komponente implementiert eine oder mehrere "tasks".

TinyOS Scheduler realisiert die Ausführungsreihenfolge der tasks (z.B. auf FIFO Basis).

Diese Vorlesung beschäftigte sich mit den technischen Aspekten von Rechensystemen. Es gab eine Einführung in Betriebssysteme und systemnahe Programmierung. Insbesondere wurden folgende Aspekte behandelt:

Modellierung von Prozessen, z.B. Petrinetze, sowie die Synchronisation von Prozessen beim Zugriff auf gemeinsame Ressourcen.

Verwaltung von Prozessen und deren Zuteilung an die CPU, um sie auszuführen.

Verwaltung des Arbeitsspeichers aus der Sicht des Betriebssystems (virtuelle Speicherverwaltung, Seitenadressierung), persistente Speicherung von Information in Dateien.

Prozesskommunikation in lokalen und verteilten Systemen.

Verwaltung der Geräte.

Sicherheit in Rechensystemen.

Der Entwurf und die Implementierung von Betriebssystemen ist komplex und aufwändig. Wichtige Aspekte sind

definierte Abstraktionen, z.B. Prozesse, Threads, Adressraum

Definition von Schnittstellen zur Bereitstellung von Operationen, z.B. Operation zur Synchronisation, persistente Speicherung von Information

Schnittstelle sollte einfach, vollständig und effizient sein.

Sicherstellen der Abgrenzung, z.B. Isolieren von Fehlern.

Verwalten der Hardware, z.B. Geräteverwaltung.

Festlegung der Paradigmen, z.B. Ausführungsparadigma, Systemaufrufschnittstelle, Speicherplatz vs. Laufzeit.

Generated by Targeteam